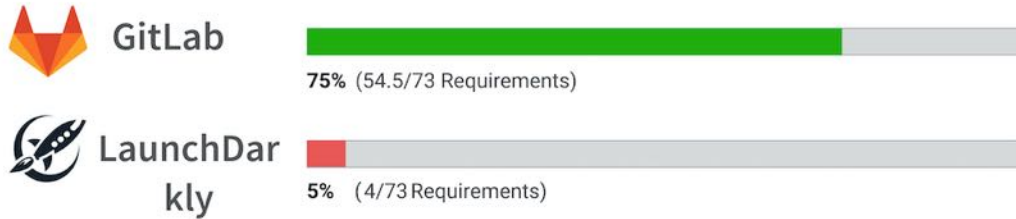


# GitLab vs LaunchDarkly

## Decision Kit



	GitLab	LaunchDarkly	Missing in LaunchDarkly
<b>Manage</b>	5.5/8	2/8	<ul style="list-style-type: none"> <li>Subgroups</li> <li>Audit Events</li> <li>Audit Reports</li> <li>Compliance Management</li> <li>Code Analytics</li> <li>DevOps Reports</li> <li>Value Stream Management Insights</li> </ul>
<b>Plan</b>	6/8		<ul style="list-style-type: none"> <li>Issue Tracking</li> <li>Kanban Boards</li> <li>Time Tracking</li> <li>Epics</li> <li>Roadmaps</li> <li>Service Desk</li> <li>Requirements Management</li> <li>Quality Management</li> </ul>
<b>Create</b>	7.5/8		<ul style="list-style-type: none"> <li>Source Code Management</li> <li>Code Review</li> <li>Wiki</li> <li>Static Site Editor</li> <li>Web IDE</li> <li>Live Preview</li> <li>Snippets</li> <li>Design Management</li> </ul>
<b>Verify</b>	6/8	1/8	<ul style="list-style-type: none"> <li>Continuous Integration</li> <li>Code Quality</li> <li>Code Testing and Coverage</li> <li>Load Testing</li> <li>Web Performance</li> <li>Usability Testing</li> <li>Accessibility Testing</li> <li>Merge Trains</li> </ul>
<b>Package</b>	4.5/6		<ul style="list-style-type: none"> <li>Package Registry</li> <li>Container Registry</li> <li>Helm Chart Registry</li> <li>Dependency Proxy</li> <li>Jupyter Notebooks</li> <li>Git LFS</li> <li>Dependency Firewall</li> </ul>
<b>Secure</b>	7/8		<ul style="list-style-type: none"> <li>SAST</li> <li>DAST</li> <li>Fuzz Testing</li> <li>Dependency Scanning</li> <li>Container Scanning</li> <li>License Compliance</li> <li>Secret Detection</li> <li>Vulnerability Management</li> </ul>
<b>Release</b>	7/8	2/8	<ul style="list-style-type: none"> <li>Continuous Delivery</li> <li>Pages</li> <li>Review Apps</li> <li>Advanced Deployments</li> <li>Feature Flags</li> <li>Release Orchestration</li> <li>Release Evidence</li> <li>Secrets Management</li> </ul>
<b>Configure</b>	4.5/7		<ul style="list-style-type: none"> <li>Auto DevOps</li> <li>Kubernetes Configuration</li> <li>ChatOps</li> <li>Runbooks</li> <li>Serverless</li> <li>Infrastructure as Code</li> <li>Cluster Cost Optimization</li> </ul>
<b>Monitor</b>	5/8		<ul style="list-style-type: none"> <li>Metrics</li> <li>Alert Management</li> <li>Incident Management</li> <li>Logging</li> <li>Tracing</li> <li>Error Tracking</li> <li>Product Analytics</li> <li>Synthetic Monitoring</li> </ul>
<b>Defend</b>	1.5/3		<ul style="list-style-type: none"> <li>Web Application Firewall</li> <li>Container Host Security</li> <li>Container Network Security</li> </ul>

## Summary

Launchdarkly is a feature delivery and feature management service offering. This means that the service they provide allows their customers to dynamically control when new features (within their application) become available to their end users. Launchdarkly's service offering represents a tool within the DevOps Toolchain that aligns within the Continuous Delivery (CD) tool category.

## Strengths

Launchdarkly provides a very sophisticated feature management service that enables Development and Operations teams to deliver code faster with minimal risk by using feature flags to deploy code when they want while keeping new features hidden until product and marketing teams are ready to share. Their service allows customers to take full control of feature releases through:

- Fast feature delivery: separating code deployments from feature deliveries

...control when new features roll out to the end user, separate code deployments from feature releases, and control when new features are available to different segments of users.

-incremental changes: controlling feature rollouts into predefined stages and slowly test the functionality within production before confirming it's safe to proceed to the next step

-Canary launches: rolling features out to a small number of users to assess the reaction of the overall system (also known as Percentage Releases)

-Dynamic feature control: coordinating and delegating release control to other teams in your organization, turning off features when they are causing performance issues or poor user experiences (feature kill switch), and retiring features.

-Automated feature delivery: setup scripts that automatically turn features off/on or adjust a flag's targeting rules based on metrics from Application Performance Management (APM) tools

-Analytics: providing deep insights across all features and how they are performing with Flag Insights; understanding how the application and team members are using LaunchDarkly.

## Gaps

Launchdarkly is a niche solution that must be overlaid within a customer's existing DevOps Toolchain. Though they do provide more comprehensive feature flag configuration capabilities, they lack:

-A Holistic DevOps Solution: unable to meet customers complete DevOps needs with a single application Toolchain

-Reasonable/Economical pricing: high per-user price point for a single tool (Launchdarkly's pricing starts at \$90/user for their lower tier plan, \$390/user for their mid-tier plan and they don't offer a free base tier)

-Industry recognition: no industry recognition by research and advisory bodies such as Gartner and Forrester.

-Meeting/Phone Support: no option to request escalating a technical support inquiry to a troubleshooting call/meeting

Gitlab's DevOps Toolchain includes **feature flags for feature management** as Launchdarkly does. Unlike Launchdarkly, Gitlab's Toolchain offers deeper features/functionalities necessary for quickly delivering high quality applications and services for the entire software lifecycle in a single application. Examples of GitLab features/functionalities that are not offered by Launchdarkly include **Source Code Management (SCM)**, **Continuous Integration (CI)** and **Security Embedded within the DevOps workflow (DevSecOps)**.

## Resources

[Launchdarkly.com](https://launchdarkly.com)

[Launchdarkly Pricing](#)

[Launchdarkly Wikipedia](#)

## Feature Comparison



### FEATURES

#### Feature Flags

CORE	STARTER	PREMIUM	ULTIMATE
FREE	BRONZE	SILVER	GOLD

This feature gives you the ability to configure and manage feature flags for your software directly in the product. Simply create a new feature flag, validate it using the simple API instructions in your software, and you have the ability to control the behavior of your software via the feature flag within GitLab itself. Feature Flag strategies can be set per environment. GitLab Feature Flags includes an API for interacting with them.



[Learn more about Feature Flags](#)

#### Feature Flag List view

CORE	STARTER	PREMIUM	ULTIMATE
FREE	BRONZE	SILVER	GOLD

This feature gives you the ability to view all the feature flags configured in a project. You can toggle the flags on or off directly from this page, and view all the associated information for a flag. This includes the strategies linked to the flag, the number or percent of users affected, and the environments.



[Learn more about Feature Flags](#)

#### Percent of Users Strategy for Feature Flags

CORE	STARTER	PREMIUM	ULTIMATE
FREE	BRONZE	SILVER	GOLD

You can select “Percent of Users” as a rollout strategy for your feature flags. This allows percentages to be set individually for each environment and each flag. When “Percent of Users” is configured and the flag is enabled, the feature will be shown to the configured percentage of logged-in users. This allows you to do controlled rollouts and monitor the behavior of the target environment to ensure the results are as expected.



[Learn more about Percentage Rollout for Feature Flags](#)

#### UserID Rollout Strategy for Feature Flags



You can choose “User ID” as a rollout strategy for your feature flags. The User ID strategy allows you to specify a comma-separated list of User IDs and then toggle a feature flag only for the specified users. This can allow you to target testing features with specific cohorts or segments of your userbase.



[Learn more about UserID Rollout for Feature Flags](#)

#### Set multiple strategies per environment



You can define multiple strategies independent of environments via API and/or UI.



[Learn more about setting Feature Flag strategies per environment](#)

#### User List Strategy for Feature Flags



You can choose “User List” as a rollout strategy for your feature flags. User lists can be reused for multiple feature flags while allowing you to manage them in a single location. You can create Feature Flag user lists from the API, and edit or delete them from the API or UI.



[Learn more about setting Feature Flag strategies per environment](#)

#### Associate Feature Flags with the issue(s) that is related to them



You can create a link from the issue that introduced the feature flag to the feature flag itself. That relationship is visible in the feature flag details.



[Learn more about linking related issues to feature flags](#)

[Get Your Free Trial](#)



### Why GitLab?

- Product
- Solutions
- Services
- DevOps lifecycle
- DevOps tools
- Is it any good?
- Releases
- Pricing

### Resources

- All resources
- All-Remote
- Blog
- Newsletter
- Events
- Webcasts
- Topics
- Training

### Community

- Customers
- Contribute
- Partners
- Channel Partners
- Explore repositories
- Source code
- Shop
- Direction

### Support

- Get help
- Contact Sales
- Contact Support
- Support options
- Status
- Customers portal

### Company

- About
- What is GitLab?
- Jobs
- Culture
- Team
- Press
- Analysts
- Handbook

[Get started](#)

[Docs](#)

[Contributors](#)

[Security](#)

[Install](#)

[Core Team](#)

[Contact](#)

[Hall of fame](#)

[Terms](#)

[Community Forum](#)

[Privacy](#)

[Trademark](#)

Git is a trademark of Software Freedom Conservancy and our use of 'GitLab' is under license

[View page source](#) — [Edit in Web IDE](#) — [please contribute.](#) 